

# 情報の科学 学年末考查

次の魚住先生と Y さんの会話を読み、問いに答えよ。

魚住先生：硬貨の枚数を数えるプログラムを作りたいと思います。

Y さん：スーパーのレジで動いているプログラムですね。でもどうやって枚数を数えるんですか？

魚住先生：例えば、752 円を硬貨の枚数が一番少なくなるように渡すとしたら、それぞれの硬貨は何枚必要ですか？

Y さん：えっと、500 円玉が【1】枚、100 円玉が【2】枚、50 円玉が【3】枚、10 円玉が【4】枚、5 円玉が【5】枚、1 円玉が【6】ですね。

魚住先生：どうやって計算しましたか？

Y さん：まず、500 円玉が何枚になるかを考えました。752 を【7】で割った【8】が枚数ですよね。

魚住先生：その通りです。じゃあ 500 円玉の分を引いた残金は、どうやって計算したら良い？

Y さん：えーっとお、あ！そうか！【9】で割った【10】がそのまま残金になる！

魚住先生：コツがつかめたようですね。ではその流れでプログラムを組んでみてください。

【1】～【6】はマークの数字で解答し、【7】～【10】は次の選択肢より解答せよ。

- ①500 ②100 ③50 ④10 ⑤5 ⑥1 ⑦商 ⑧余り

Y さん：先生、とりあえずプログラムが書けました。500 円の部分だけですけど。

```
kingaku = 752
goukei = 0
zankin = kingaku
maisu = zankin 【14】 500
zankin = zankin % 【11】
print("500 円玉は", 【12】 , "枚")
```

【11】～【12】に入る内容を選択せよ。(2×2 点)

- ①kingaku ②goukei ③zankin ④maisu ⑤500 ⑥100 ⑦50 ⑧10 ⑨5 ⑩1

魚住先生：上出来です。この変数 goukei は何のために 0 処理をしたんですか？

Y さん：1 円玉の枚数まで数えたら、最後に硬貨の枚数を合計したくて付けました。

魚住先生：最高かよ・・・。でもそれだと、合計を計算する部分が 1 行足りませんね。

Y さん：どこに追加したら良いですか？

魚住先生：5 行目あたりに追加すると良いですね。

【13】先のことを考え、硬貨の枚数を合計するための計算式を選択せよ。

①goukei = goukei + zankin

②goukei = goukei + kingaku

③goukei = kingaku - zankin

④goukei = goukei + 1

⑤goukei = goukei + 500

⑥goukei = goukei + maisu

Y さん：先生、書けました。

```
kingaku = 752
goukei = 0
zankin = kingaku
maisu = zankin 【14】 【15】
【13】
zankin = zankin % 【11】
print("500 円玉は",mais,"枚")
maisu = zankin 【14】 【16】
【13】
zankin = zankin % 100
print("100 円玉は",mais,"枚")
maisu = zankin 【14】 【17】
【13】
zankin = zankin % 50
print("50 円玉は",mais,"枚")
maisu = zankin 【14】 【18】
【13】
zankin = zankin % 10
print("10 円玉は",mais,"枚")
maisu = zankin 【14】 【19】
【13】
zankin = zankin % 5
print("5 円玉は",mais,"枚")
maisu = zankin 【14】 【20】
【13】
zankin = zankin % 1
print("1 円玉は",mais,"枚")
print(kingaku,"円は全部で", 【21】 , "枚")
```

【14】に入る内容を選択せよ。

①+ ②- ③\* ④/ ⑤// ⑥% ⑦\*\*

【15】～【21】に入る内容を選択せよ。

①kingaku ②goukei ③zankin ④maisu ⑤500 ⑥100 ⑦50 ⑧10 ⑨5 ⑩1

魚住先生：正しく動いたようだね。でもこのプログラム、同じことを何度も繰り返しているよね。

Yさん：そうですねえ。毎回同じように割ったり余りを出したりしていますね。これってまとめられないんですか？

魚住先生：できるで。硬貨の種類をリストに入れたら、繰り返していけるはずだ。

```
kingaku = int(input())
kinshu=[500,100,50,10,5,1]
goukei = 0
maisu=[0,0,0,0,0,0]
zankin = kingaku
for i in range(len(kinshu)):
    maisu[i] = 【22】 // 【23】
    goukei = 【24】 + 【25】
    zankin = 【26】 % 【27】
for i in range(len(kinshu)-1,-1,-1):
    print( 【28】 ,"円玉は", 【29】 ,"枚")
print( 【30】 ,"円は全部で", 【31】 ,"枚")
```

【22】～【31】に入る内容を選択せよ。

①maisu[i] ②goukei ③zankin ④kinshu ⑤kinshu[a] ⑥kingaku ⑦kinshu[i] ⑧kinshu[j]

Yさん：できました。かなりスッキリしました。

魚住先生：よくここまで省略できたね。素晴らしい。良いプログラムというのは、少ない行数で同じように動くもののことを言うんですよ。ちなみにこのプログラムで、kingaku=987 だったら何と表示されるかやってみてくれる？

Yさん：kingaku=987 ですね、やってみたらこんな感じに表示されました。

```
【32】 円玉は 【38】 枚
【33】 円玉は 【39】 枚
【34】 円玉は 【40】 枚
【35】 円玉は 【41】 枚
【36】 円玉は 【42】 枚
【37】 円玉は 【43】 枚
【44】 【45】 【46】 円は全部で 【47】 枚
```

【32】～【37】に入る内容を選択せよ。

①kingaku ②goukei ③zankin ④maisu ⑤500 ⑥100 ⑦50 ⑧10 ⑨5 ⑩1

【38】～【46】は当てはまる数値の番号をマークせよ。

【47】は次から選択せよ。①5 ②6 ③7 ④8 ⑤9 ⑥10 ⑦11 ⑧12 ⑨13 ⑩14

Yさん：魚住先生、もっと簡単な問題は無いんですか？

魚住先生：簡単？これまでの問題はどれも簡単だと思うけど？仕方ないな。じゃあ、頭の体操として、九九の1の段から9の段までを一気に出力するプログラムを書いてみてよ。

Yさん：あ、確かに九九なら簡単そう。二重ループでサクッと書けました。

```
for i in range(9):
    for j in range(9):
        print(( i 【48】 【49】 ) 【50】 ( j 【51】 【52】 ))
```

【48】～【52】に入る内容を選択せよ。(2×5点)

①+ ②- ③\* ④/ ⑤// ⑥% ⑦\*\* ⑧0 ⑨1 ⑩2

魚住先生：では次に、リストa[0]からa[i]までの和が100未満となる最大のiを探しましょう。

Yさん：100未満になるまで足していくと、どこまで足せますか？ってことですよ？

魚住先生：そう。それをコンピュータに探させます。

```
a = [24,19,33,15,21,18,42]
s = 0
i = 0
①
while 【53】 + 【54】 < 100:
    ②
    s = 【55】 + 【56】
    ③
    i = 【57】 + 1
④
print("a[0]からa[" ,i, "]までの和は", 【58】 )
```

【53】～【58】に入る内容を選択せよ。(2×6点)

①a ②s ③i ④a[i] ⑤a[s]

※ 【53】 と 【54】 は順不同とする

Y さん：なんとか動いたと思います。

a[0]からa[ 4 ]までの和は 91

魚住先生：あれ？a[4]までだと、【59】【60】【61】になって、100 を超えてしまうよね？

Y さん：本当だ。でも 91 って出ているので、a[3]までの和が表示されているんですよ。

魚住先生：困ったな。それじゃ、あまり美しくないけど、「i = i - 1」を【62】に加えて、帳尻を合わせようか。

【59】【60】【61】に入る数値をそれぞれマークせよ。

【62】は、「i = i - 1」コード内のどこに加えると良いか、①～④から選択せよ。

Y さん：先生、流石に少し疲れてきました。

魚住先生：じゃあここで少し、授業の復習をしましょうか。探索についておさらいしましょう。

Y さん：(順次構造からじゃないんかい)探索って、リストの中から数値を探すやつでしたよね。

魚住先生：逐次探索と二分探索について学習しましたね。復習として、次の問題を考えてください。

「16, 23, 31, 48, 52, 65, 79」の中から「16」「23」「52」「79」を探索するとき、逐次探索と二分探索のそれぞれのアルゴリズムでの探索回数を答えよ。

データ	逐次探索での回数	二分探索での回数
16	【63】	【67】
23	【64】	【68】
52	【65】	【69】
79	【66】	【70】

```
list = [16,23,31,48,52,65,79]
i = 0
j = 【71】
s = 36
count = 0
while i<=j:
    count+=1
    m = int( 【72】 )
    if list[ 【73】 ]==s:
        print("Hakken!",count)
        break
    if list[m] < s:
        【74】
    else:
        【75】
```

二分探索のプログラムを左下の通り組んでみた。

【71】に入る数値をマークせよ。

【72】～【75】に入る内容を選択せよ。

- ① i-j    ② i+j/2    ③ (i+j)/2    ④ i = m + 1    ⑤ i = m - 1    ⑥ j = m + 1    ⑦ j = m - 1  
⑧ i        ⑨ j        ⑩ m

左下の二分探索のプログラムにおいて、「s=52」と修正した。この時、「52」が見つかるまでの変数 i,j,m および list[m]の値を答えよ。

i の値	j の値	m の値	list[m]の値
0	6	3	48
【76】	【77】	【78】	【79】
【80】	【81】	【82】	【83】

【79】と【83】は次から選択せよ。    ①16 ②23 ③31 ④48 ⑤52 ⑥65 ⑦79

Y さん：授業の時はプログラムを組んだだけだったので、ここまで細かくトレースしたことはありませんでした。良い勉強になりました。

魚住先生：では最後に、うるう年判定プログラムも復習しましょうか。

```
seireki = int(input())
①
if seireki%4==0:
    uruu = 1
    ②
    if seireki%100==0:
        uruu = 0
        ③
        if seireki%400==0:
            uruu = 1
            ④
if uruu == 1:
    print("うるう年です")
else:
    print("うるう年ではありません")
```

Y さん：これを作ってみたんですが、「2021」と入力するとエラーが出てしまいます。

魚住先生：あー、これは、【84】を【85】に追加すると動きますよ。

【84】に入る内容を選択せよ。(2点)

- ①uruu=1 ②uruu=0 ③else: ④elif: ⑤seireki=2021 ⑥seireki=0 ⑦if seireki%4!=0:

【85】は【84】をプログラムに追加する場所を①～④で答えよ。(2点)

Y さん：あ！本当だ！「うるう年ではありません」って表示されました！ありがとうございました！

魚住先生：いえあ